

Amendments to the Specification:

Please replace the paragraph on page 2, lines 6-15 with the following amended paragraph:

A solution known in the art for monitoring performance of distributed applications is provided by the Application Response Measurement (ARM) standard as described in “The Application Response Measurement (ARM) API, Version 2”, Mark W. Johnson, Tivoli Systems, December 1997 (~~available at~~ <http://regions.emg.org/regions/emgarmw/index.html>). The standard defines some API calls, which can be used to ask an agent to measure transactions and to make the information available to management applications. In this way, an accurate picture of the actual workload of the system can be obtained.

Please replace the paragraph on page 3, lines 17-29 with the following amended paragraph:

A different solution for monitoring performance of distributed applications where source code changes are not possible is described in “Service management using the application response measurement API without application source code modification”, Martin Haworth, Resource and Performance Management Solutions Network and System Management Division, Hewlett-Packard Company, June 1997 (~~on~~ <http://regions.emg.org/regions/emgarmw/index.html>). This article proposes capturing a script by recording the user actions on the client, for example, by means of a Remote Terminal Emulation (RTE) technique. The user script is edited to include calls to the ARM APIs for each desired transaction. The user script can then be scheduled to run at an appropriate interval.

Please replace the two paragraphs extending from page 10, line 35 to page 11, line 15 with the following amended paragraphs:

Considering now Figures 3a-3d, a process 300 corresponding to the execution of a transaction begins at the black start circle 302 in the swim-lane of the browser. Proceeding to block [303] 304, a web page requested by the user is downloaded on the client; the web page is then displayed on the monitor of the client at block 304. As soon as the user selects a link on the current web page (for example, clicking with the mouse on its graphical representation), a corresponding HTTP request for the corresponding server is generated ~~at block 305~~.

The HTTP request is intercepted by [the] a sensor at block 306. ~~The process then branches at block 307,~~ wherein a test is made to determine whether the HTML code defining the current web page includes a predetermined keyword for enabling the monitoring of the transactions originating therefrom.

If the keyword has been found, a correlation flag is asserted at block 308; conversely, the correlation flag is disasserted at block 310. In both cases, the process merges again at block 312.

Please replace the paragraph on page 12, lines 21-32 with the following amended paragraph:

In any case, the process then continues to block 344, wherein the query is called on the application (possibly passing the correlator). In response thereto, the process proceeds to block [346] 345 in the swim-lane of the application; assuming that the query involves the execution of a further sub-transaction on an auxiliary server, the application generates a corresponding HTTP request. A test is then made at decision block 348 to verify whether the correlator has been passed to the application in the call. If so, the process enters block 350 wherein the auxiliary HTTP request is updated adding the correlator; the process then continues to block 352. Conversely, the flow of activities descends into block 352 directly.

Please replace the Abstract paragraph on page 28, lines 5-18 as was amended in the Preliminary Amendment filed October 16, 2003 with the following further amended Abstract paragraph:

~~A method~~ Method and ~~a corresponding~~ system for performance monitoring ~~performance~~ of distributed applications ~~are proposed. In the method of the invention, a~~ A sensor intercepts every request of service for a server that is generated on a client. If the request meets a filtering condition (for example, defined by the address of the server, the web page from which the request is originated and/or the selected link) the measuring of a corresponding transaction on the client is enabled; at the same time, the request is updated by inserting a correlator. The request is then transmitted to the server. If the request includes the correlator, the measuring of a sub-transaction originating from the request is also enabled on the server. The parameters measured on the client and on the server are then associated with the correlator.